ATME COLLEGE OF ENGINEERING

13th KM Stone, Bannur Road, Mysore - 560 028

ATME

College of Engineering

DEPARTMENT OF COMPUTER SCIENCE AND DESIGN

LECTURE NOTES

COURSE: MULTIMEDIA SYSTEM AND DESIGN

COURSE CODE: BCG613A

SEMESTER: VI

COURSE COORDINATOR: Mrs.DIVYA N

(ACADEMIC YEAR 2025-26)

MULTIMEDIA SYSTEM DESIGN-BCG613A

MODULE-2 NOTES

TEXT

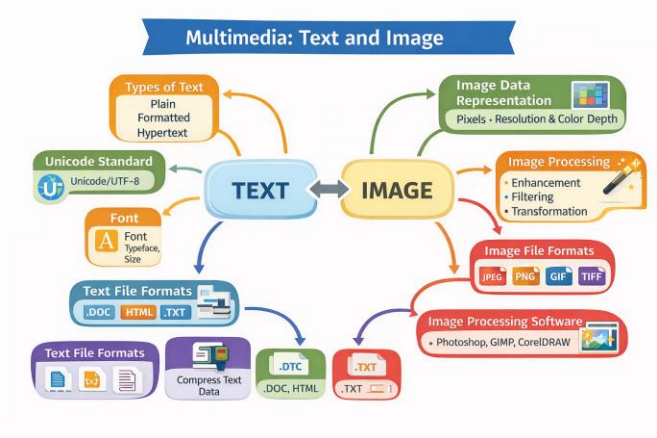
Introduction

Text is the most basic and widely used component of multimedia. In multimedia systems, text is not just plain written content but also includes styled and formatted content that helps convey information clearly. Text provides the foundation for menus, labels, instructions, descriptions, titles, and captions. It plays a crucial role in organizing information and guiding users through multimedia presentations.

Text is a collection of characters such as letters, numbers, symbols, and special characters used to represent information in a readable and meaningful form. In multimedia systems, text is the basic and simplest form of data used to convey information, and it can be stored, processed, and transmitted in digital form using encoding standards like ASCII or Unicode.

There are three types of text used in multimedia:

1. unformatted text
2. formatted text,
3. hypertext.



1. Plain Text

Plain text consists of a simple sequence of characters without any formatting, such as bold, italics, or colors. It is stored in basic text files, usually with a .txt extension, and can be read by any text editor. Plain text is lightweight and widely compatible across platforms, but it does not support styling or multimedia integration. Despite its simplicity, plain text is often used for instructions, notes, or as input for multimedia applications where formatting is not required.

2. Formatted Text

Formatted text includes style attributes such as font type, size, color, alignment, and emphasis like bold or italic. This type of text is used in word processors, presentations, and multimedia applications where appearance and readability are important. Formatted text allows designers to highlight important information, structure content logically, and improve the visual appeal of multimedia. For example, headings, subheadings, and highlighted text in a presentation are all instances of formatted text.

3. Hypertext

Hypertext is text that contains **links** to other pieces of information, such as documents, web pages, or multimedia content. It is the backbone of interactive systems like websites, e-learning modules, and multimedia applications, enabling users to navigate non-linearly through content. Hypertext allows users to click on links and jump to related topics, enhancing interactivity and improving access to information. This type of text is essential in web-based multimedia and hypermedia systems.

Architecture of a Hypertext Document

Hypertext system has **three layers**:

1. Presentation Layer

- User interface layer
- Displays text and images
- Handles user interactions (mouse clicks)

2. Hypertext Abstract Machine (HAM)

- Middle layer
- Knows structure of document
- Manages nodes and links
- Least system dependent

3. Storage Layer

- Database layer
- Stores data
- Manages transactions and data properties

Unicode Standard

The **Unicode Standard** is a universal character encoding system designed to represent text from all the world's writing systems in a single, consistent format. Prior to Unicode, different computer systems and software used multiple character sets, which often led to **incompatibility and incorrect display of text**, especially for multilingual content. Unicode solves this problem by assigning a **unique numeric code (code point)** to every character, symbol, or punctuation mark in almost all languages, including Latin, Cyrillic, Arabic, Chinese, and special symbols.

This standard ensures that text is encoded, stored, transmitted, and displayed consistently across different platforms, software, and devices. For example, the Unicode code point U+0041 always represent the capital letter 'A', regardless of the system or application being used. Unicode supports more than 143,000 characters and continues to expand with new scripts and symbols, making it essential for global multimedia applications, websites, e-learning modules, and international software.

Unicode also enables **multilingual text processing**, allowing different languages to coexist within the same document or application. It supports **various encoding forms** such as UTF-8, UTF-16, and UTF-32, which differ in storage size and compatibility.

i. **UTF-8 (Unicode Transformation Format – 8 bit):** UTF-8 is a variable-length Unicode encoding format that uses 1 to 4 bytes to represent each character. It is backward compatible with ASCII, meaning the first 128 characters are stored using a single byte exactly as in ASCII. Characters beyond ASCII, such as accented letters, Asian scripts, and emojis, are stored using 2, 3, or 4 bytes. UTF-8 is highly memory efficient for English text and does not have byte-order issues.

ii. **UTF-16 (Unicode Transformation Format – 16 bit):** UTF-16 is a variable-length encoding format that typically uses 2 bytes to represent most common characters. Characters in the Basic Multilingual Plane (BMP) are stored using one 16-bit unit (2 bytes). However, characters outside the BMP, such as certain rare symbols and emojis, are represented using two 16-bit units called surrogate pairs (total 4 bytes). UTF-16 is efficient for languages that use large character sets, such as Chinese, Japanese, and Korean. It is widely used in systems like Windows operating system, Java, and .NET environments.

iii. **UTF-32 (Unicode Transformation Format – 32 bit):** UTF-32 is a fixed-length encoding format that uses exactly 4 bytes for every character. Each Unicode code point is directly stored in a 32-bit unit, making character processing and indexing very simple. Since every character occupies the same amount of space, there is no need for surrogate pairs. However, UTF-32 consumes more memory compared to UTF-8 and UTF-16, making it less efficient for storage and transmission. It is mainly used in internal system processing where simplicity is more important than memory efficiency.

Font

A **font** refers to a particular design of text characters, including style (e.g., bold, italic), weight, and size. Fonts determine how text appears visually on the screen or in print. In multimedia, choosing appropriate fonts is important for readability, aesthetics, and user experience. Fonts can be system-based or embedded within multimedia applications**.**

Font appearance describes the overall visual look of the text based on its design characteristics. It may appear formal, modern, decorative, or simple. For instance, serif fonts like Times New Roman give a traditional and formal appearance, while sans-serif fonts like Arial provide a clean and modern look. Decorative fonts are often used in posters and invitations to create an attractive design.

Font size refers to how large or small the text appears and is measured in points (pt), where 1 point equals 1/72 of an inch. For example, 12 pt is commonly used for normal paragraph text, 14 pt or 16 pt for subheadings, and 18 pt or above for titles. Proper font size ensures good readability in both printed and digital documents.

Font style refers to special formatting applied to text to emphasize important information. Common styles include bold, italic, underline, and strikethrough. For example, bold text is used for headings, italic text is used for emphasis or quotations, underline is used to highlight important points, and strikethrough is used to show deleted content. These styles help in improving clarity and visual presentation of the text.

Text compression is the process of reducing the size of text data so that it occupies less storage space and can be transmitted more efficiently over networks. In multimedia systems, text compression is important because even though text files are smaller than images or audio, large amounts of text such as e-books, subtitles, or logs can still consume significant storage and bandwidth. Compression ensures that data transfer is faster and storage requirements are minimized without compromising information.

There are two main types of text compression: **lossless** and **lossy**, though for text, **lossless compression** is almost always used because it is essential that the original text can be perfectly reconstructed. Lossless compression algorithms analyze patterns in the text and replace repetitive or predictable sequences with shorter codes.

HuffmanCoding:
Huffman coding is a lossless text compression technique based on the frequency of characters. In this method, characters that occur more frequently in the text are assigned shorter binary codes, while less frequent characters are assigned longer codes. The process involves calculating the frequency of each character, constructing a binary tree called a Huffman tree, and then generating unique prefix codes for each character. Since no code is a prefix of another, decoding becomes easy and error-free. Huffman coding reduces the overall number of bits required to represent the text and is widely used in file compression formats.

Lempel–Ziv(LZ77)Coding:
Lempel–Ziv coding, particularly LZ77, is a dictionary-based lossless compression technique that works by replacing repeated occurrences of text with references to previous occurrences. It uses a sliding window mechanism to search for matching strings in already processed data. Instead of storing repeated text again, it stores a pointer containing information such as distance and length of the match. This method efficiently removes redundancy in text containing repeated patterns and forms the basis of many modern compression algorithms.

Lempel–Ziv–Welch(LZW)Coding:
Lempel–Ziv–Welch (LZW) coding is an improved version of the Lempel–Ziv method that builds a dictionary of repeated character sequences dynamically during compression. Initially, the dictionary contains all possible single characters, and as the text is processed, new patterns are added to the dictionary. Instead of transmitting the actual repeated strings, the encoder sends dictionary indices corresponding to those patterns. LZW provides better compression efficiency for text with recurring patterns and is widely used in formats such as GIF and some file compression utilities.

Text compression also improves the efficiency of multimedia applications that use subtitles, captions, or textual metadata embedded in videos, presentations, or e-learning content. By compressing text data, developers can **save bandwidth**, **speed up downloads**, and reduce **memory usage** without affecting readability. Moreover, compressed text can be decompressed in real-time, allowing multimedia systems to deliver interactive content smoothly.

Text File Formats

Text file formats determine how text data is **stored, organized, and interpreted** by multimedia systems and software applications. Choosing the right text file format is important because different formats provide varying levels of **styling, structure, and metadata support**, which directly affect how text appears and can be manipulated within multimedia projects.

1. **PlainText(.txt):**
   Plain text files contain only raw characters without any formatting, fonts, or styles. They are lightweight and universally compatible, making them ideal for storing simple instructions, configuration files, or notes. However, plain text cannot support rich formatting or multimedia integration, which limits its use in visually-rich applications.

2. **WordProcessorFiles(.doc,.docx):**
   Files created by word processors such as Microsoft Word or LibreOffice Writer allow **rich text formatting**, including different fonts, sizes, styles, colors, bullet points, tables, and embedded images. These formats are widely used in multimedia for creating documents, reports, and interactive e-learning materials. They provide a combination of textual content and layout information, which is critical when precise formatting and presentation are required.

3. **RichTextFormat(.rtf):**
   RTF is a cross-platform format designed to store text with basic formatting such as bold,

italics, font sizes, and colors. Unlike proprietary word processor files, RTF can be opened and edited across different operating systems and software. It is often used in multimedia applications. where some styling is needed, but full word processor features or embedded objects are not required.

4. **MarkupLanguageFiles(.html,.xml):**
   Markup files store text along with **structural and semantic tags** that define how the content should be displayed or processed. HTML files are used for web-based multimedia, allowing integration of text with images, links, audio, video, and interactive elements. XML files, on the other hand, focus on structuring and describing data, enabling text to be interpreted by different software systems consistently. Markup files are essential in multimedia for **hypertext navigation, data exchange, and interactive content**.

5. **PostScript** is a page description language used to describe the layout and appearance of text and graphics for printing. It was developed by Adobe Inc. in 1982. PostScript is not just a simple text format; it is a programming language that tells printers exactly how to display text, fonts, images, and graphics on a page.

6. **OtherSpecializedTextFormats:**
   Some multimedia applications use specialized text formats for **subtitles (.srt, .sub)**, **closed captions**, or **metadata files** that provide additional information about media content. These formats may include timestamps, styling, and language codes to synchronize text with audio or video in multimedia presentations.

IMAGE

Introduction

An image in multimedia refers to a visual representation such as a photograph, drawing, or picture. Images play a significant role in enhancing understanding, adding realism, and improving aesthetic appeal in multimedia systems. They can be scanned, captured by cameras, or generated by software, and are processed and stored in digital form for use in applications.

Image Data Representation

Image data representation involves converting real-world visual information into digital data that a computer can store and process. In digital form, an image is represented as a grid of tiny units called **pixels (picture elements)**, where each pixel has a specific color and intensity value. The representation depends on factors such as **color depth** (number of bits per pixel) and **resolution** (number of pixels in width and height). Higher color depth and resolution result in better image quality but require more storage.

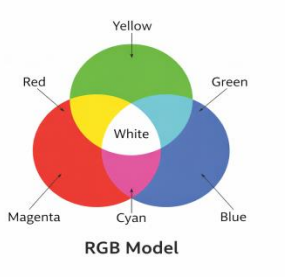
Image Processing

Definition:
Image processing is the **technique of performing operations on digital images** to enhance their quality or extract useful information. It allows multimedia designers and engineers to improve image clarity, correct defects, and prepare images for display, analysis, or further multimedia integration.

Objectives of Image Processing:

1. **Enhancement:** Improving the visual quality of an image for human interpretation, e.g., adjusting brightness, contrast, or color balance.
2. **Restoration:** Removing distortions, noise, or blurring caused during image capture or transmission.
3. **Analysis & Recognition:** Extracting information such as shapes, edges, or patterns for computer vision, medical diagnostics, or scientific research.
4. **Compression:** Reducing the storage size without losing significant quality for efficient storage and transmission.

Color model:



RGB Model

The RGB color model is an additive color model used for representing and displaying images in digital systems such as computers, televisions, cameras, and mobile screens. RGB stands for Red, Green, and Blue, which are the three primary colors of light. In this model, colors are produced by combining different intensities of these three primary colors. The diagram shows three overlapping circles representing Red, Green, and Blue. When two primary colors overlap, they form secondary colors: Red and Green combine to produce Yellow, Green and Blue produce Cyan, and Blue and Red produce Magenta.

When all three colors overlap equally at the center, they produce White, which indicates maximum intensity of light. If no color is present (R = 0, G = 0, B = 0), the result is Black. In digital images, each pixel is represented by three values corresponding to Red, Green, and Blue, typically ranging from 0 to 255 in an 8-bit system. Since it is an additive model, increasing the intensity of light makes the color brighter. The RGB model is widely used in image processing because digital display devices generate colors by emitting light using these three components.

CMYK model

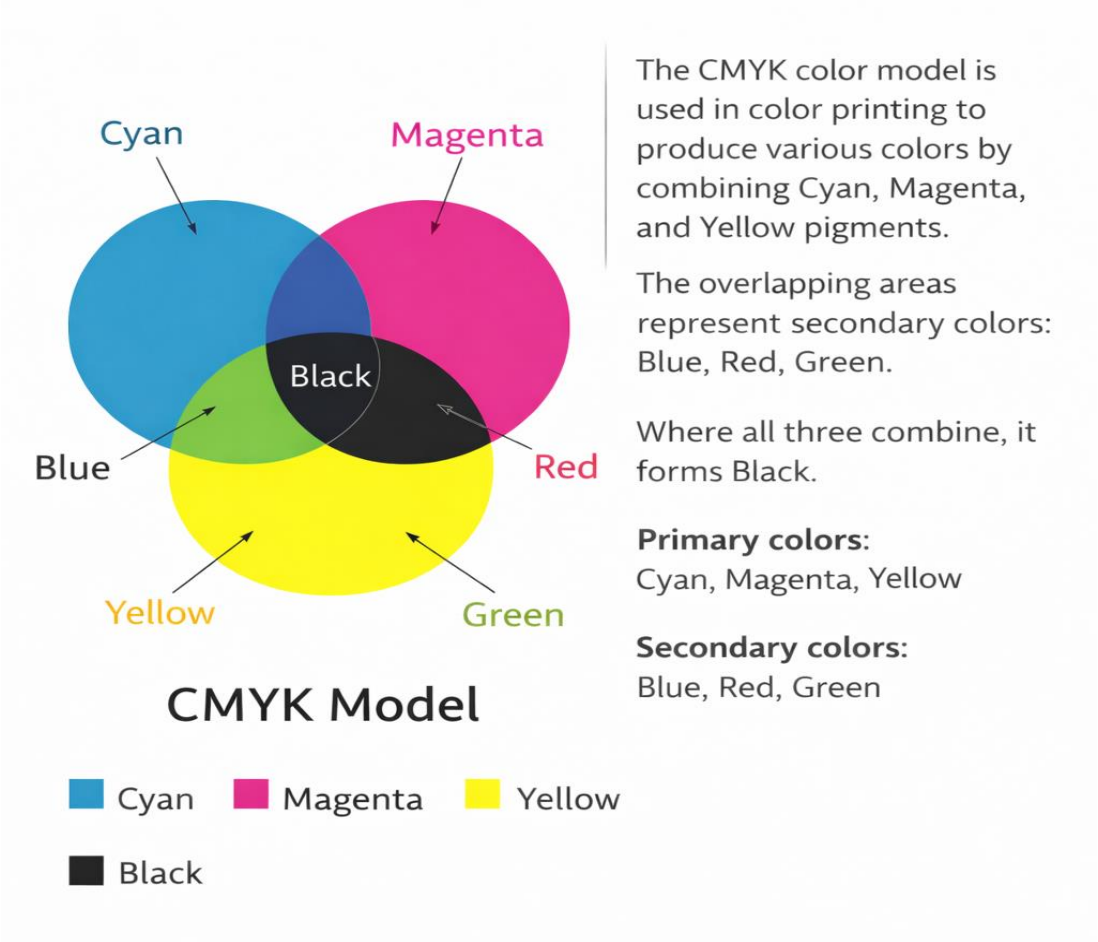


Image Types

Images can be classified based on how they are stored, displayed, and printed. The major classifications include **hard copy vs soft copy images** and **continuous tone, halftone, and bitone images**.

Hard Copy vs Soft Copy Images

A **soft copy image** is a digital image that is viewed on electronic devices such as computers, mobile phones, or televisions. It exists in digital form and is stored in formats like JPEG, PNG, or BMP. Soft copy images are displayed using light emitted from screens, usually based on the RGB color model. These images can be easily edited, copied, and transmitted over networks.

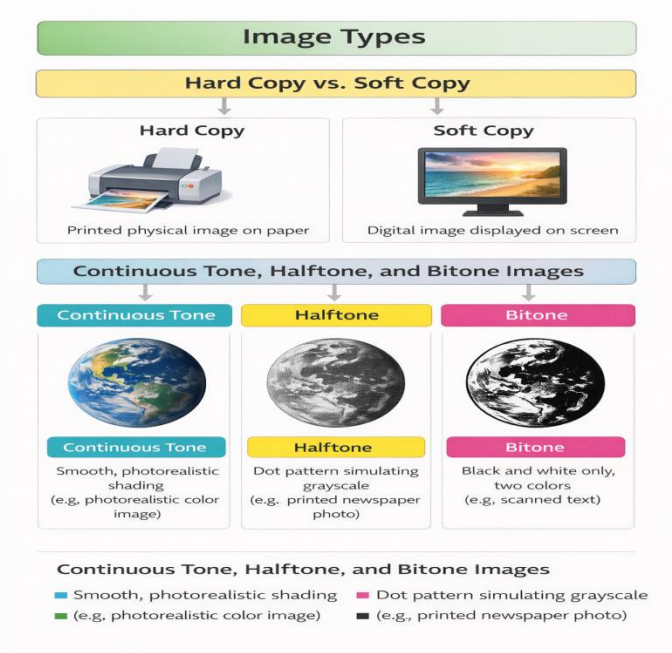
A **hard copy image**, on the other hand, is a physical printed version of an image on paper, film, or any tangible material. It is produced using printers and follows the CMYK color model in color printing. Unlike soft copies, hard copies cannot be easily modified and require physical storage spac

A **continuous tone image** contains a smooth and gradual variation of intensity and color without visible breaks between shades. These images closely resemble real-world scenes and are commonly captured using digital cameras or scanners. In continuous tone images, pixel values can vary continuously over a wide range, such as 0 to 255 in grayscale images. Photographs are typical examples of continuous tone images.

Halftone Image

A **halftone image** is created to simulate continuous tone images using small dots of varying sizes or spacing. This technique is mainly used in printing newspapers and magazines. Although the image may appear to have smooth shading from a distance, it is actually composed of tiny dots that create the illusion of varying intensity. Halftoning converts continuous tone images into printable dot patterns.

A **bitone image**, also called a binary image, contains only two possible intensity values: black and white. Each pixel is represented by a single bit (0 or 1). There are no gray levels in bitone images. They are commonly used for text documents, line drawings, and simple graphics where only two colors are sufficient. Bitone images require less storage compared to grayscale or color images.



Basic Steps of Image Processing

Image processing is the technique of performing operations on an image to improve its quality or to extract useful information. The basic steps involved in image processing are

1. **Input**
2. **Editing (Processing)**
3. **Output**.

Input Stage

The first step in image processing is acquiring or importing the image. The image may be captured using devices such as a digital camera, scanner, or satellite sensor, or it may be loaded from storage in formats like JPEG, PNG, or BMP. During this stage, the image is converted into a digital form so that it can be processed by a computer. The input image consists of pixels, and each pixel contains intensity or color values (for example, RGB values in a color image). This stage is also called image acquisition.

Example:
Scanning a photograph or loading an image file into MATLAB using imread().

Editing (Processing) Stage

The editing or processing stage involves applying various operations to modify or enhance the image. This step may include operations such as noise removal, contrast enhancement, resizing, cropping, filtering, edge detection, color conversion (RGB to grayscale), or compression. The main objective of this stage is either to improve the visual quality of the image or to extract meaningful information from it. Image processing techniques are applied mathematically to pixel values.

Example:

- Converting a color image to grayscale
- Removing noise using a filter

Output Stage

The final step is producing the processed image as output. The output may be displayed on a screen (soft copy), stored in a file, printed (hard copy), or used for further analysis. The output image is the result of all processing operations applied in the editing stage. It may be improved visually or contain extracted features useful for applications such as medical imaging, object detection, or multimedia systems.

Example:
Saving the processed image using imwrite() or displaying it on the screen.

Common Image Processing Operations:

- **Brightness and Contrast Adjustment:** Corrects underexposed or overexposed images and improves visibility of details.
- **Filtering:** Applies techniques such as smoothing, sharpening, or edge detection to highlight or suppress specific features.

- **Noise Removal:** Eliminates random variations in pixel values caused by sensor errors or transmission issues.
- **Edge Detection:** Identifies boundaries of objects within an image for recognition or segmentation.
- **Scaling and Resizing:** Changes image dimensions to fit different screen sizes or storage requirements.
- **Color Correction:** Adjusts the color balance, saturation, and hue for accurate representation or artistic effect.

Applications:

- **Medical Imaging:** Enhancing X-rays, MRI, or CT images for accurate diagnosis.
- **Satellite Imagery:** Processing Earth observation images for mapping, weather forecasting, and resource monitoring.
- **Computer Vision:** Object detection, facial recognition, and motion tracking in AI applications.
- **Photography & Multimedia:** Editing and enhancing images for advertisements, films, and presentations.

Image File Formats

Digital images are stored in **different file formats**, each optimized for specific needs such as quality, compression, or compatibility. The choice of file format depends on whether the goal is **high-quality storage, web display, transparency, or animation**.

1. **BMP (Bitmap):**
    - Uncompressed format storing every pixel individually.
    - Produces high-quality images with exact color representation.
    - Large file size makes it less suitable for internet or storage-heavy applications.
    - Often used in applications where image quality is more important than storage efficiency.
2. **JPEG/JPG (Joint Photographic Experts Group):**
    - Uses **lossy compression**, which reduces file size by discarding some image details that are less noticeable to human eyes.
    - Ideal for photographs and realistic images where slight quality loss is acceptable.
    - Widely used for web images, digital photography, and multimedia presentations.
    - Compression can be adjusted to balance between quality and file size.
3. **PNG (Portable Network Graphics):**
    - Supports **lossless compression**, preserving all image data without quality loss.
    - Supports **transparency and alpha channels**, making it ideal for graphics, logos, and web images.
    - File size is larger than JPEG for photos but smaller than BMP.
    - Preferred in multimedia applications requiring clear edges and overlays.
4. **GIF (Graphics Interchange Format):**
    - Limited to **256 colors**, making it suitable for simple graphics rather than detailed photos.

- Supports **simple animation**, making it popular for web banners and icons.
- Uses **lossless compression**, preserving exact color values within the limited palette.
- Not ideal for high-quality photographs due to color limitations.

5. **TIFF (Tagged Image File Format):**
   - High-quality, flexible format often used in **publishing, scanning, and professional imaging**.
   - Supports **lossless compression**, layers, and multiple pages.
   - Large file sizes are acceptable in professional contexts where image fidelity is critical.

Image Processing Software

Definition:
Image processing software refers to programs that provide tools to **create, edit, enhance, and manipulate digital images**. These applications are an essential part of multimedia systems because they allow designers to improve image quality, integrate images into multimedia projects, and perform complex transformations.

- Allows multimedia designers to **enhance visual content** for better communication and presentation.
- Provides **interactive editing capabilities**, making it easy to correct errors or adjust images without affecting the original file.
- Supports **integration of images** with other media elements like text, audio, and video in multimedia projects.
- Helps in professional applications like **advertising, publishing, medical imaging, scientific research**, and web design.

Key Features of Image Processing Software:

1. **Layering:** Enables multiple images or effects to be stacked on top of each other, allowing non-destructive editing.
2. **Masking:** Allows selective editing of certain areas of an image while protecting other parts.
3. **Color Correction:** Adjusts hue, saturation, brightness, and contrast for realistic or artistic effects.
4. **Transformation:** Includes scaling, rotation, cropping, and perspective adjustment to modify image shape or size.
5. **Filters and Effects:** Applies predefined effects such as blurring, sharpening, edge detection, and artistic styles.
6. **Retouching and Restoration:** Corrects image imperfections such as scratches, noise, or unwanted elements.
7. **Annotation and Metadata:** Adds text, labels, or metadata to images for documentation or interactive purposes.